



嵌入式系统

Embedded System

张武明

杭州 • 浙江大学 • 2021

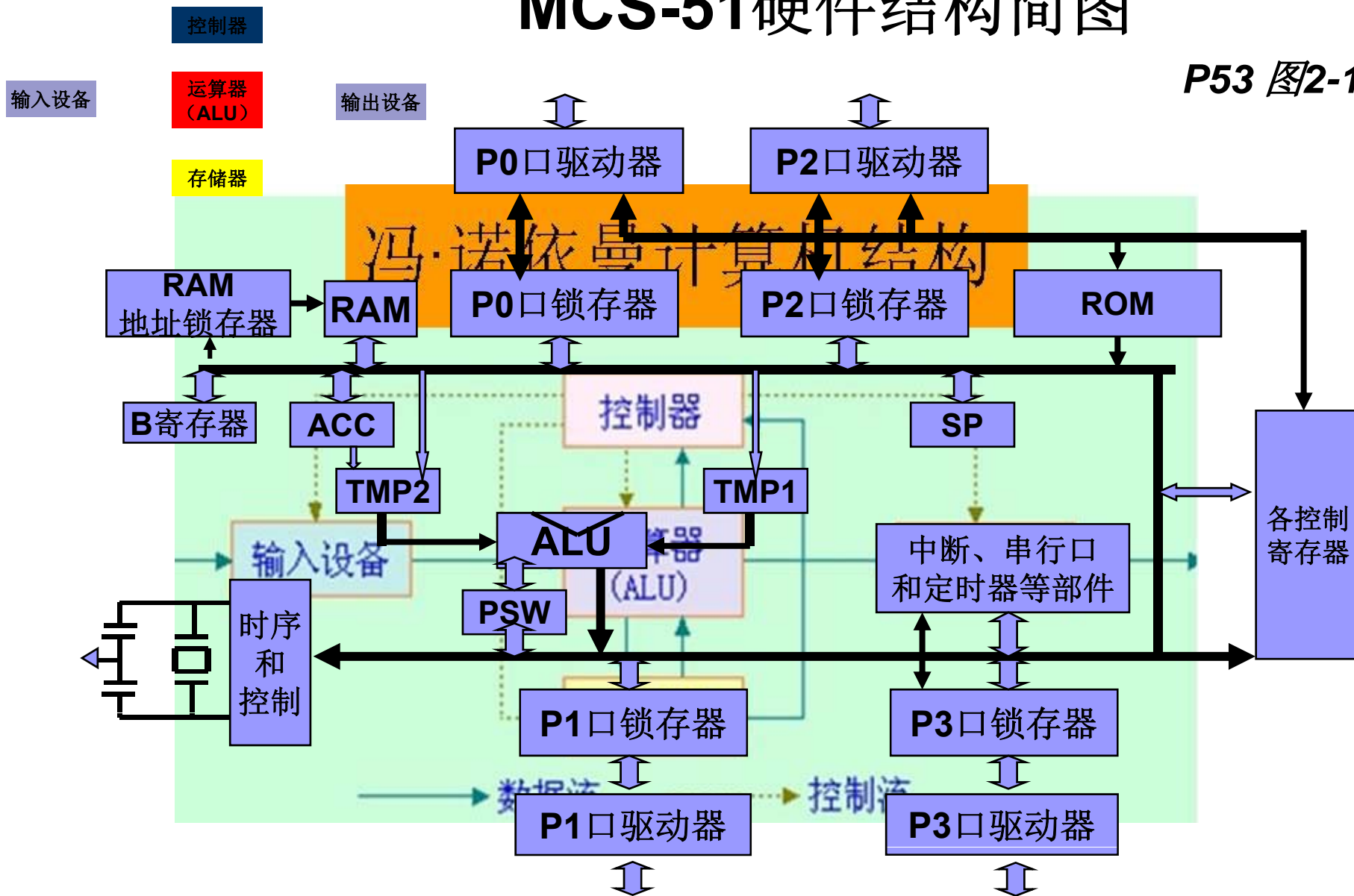


第八讲 定时器/计数器

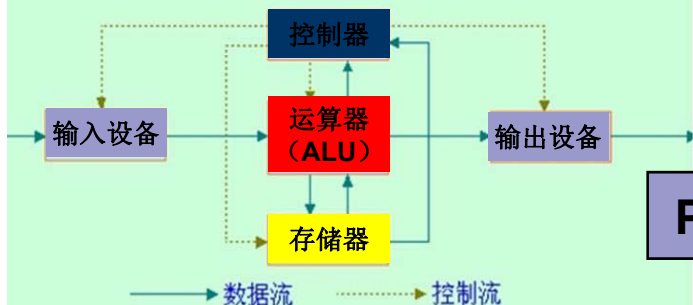
- § 8-1 定时器/计数器概述
- § 8-2 MCS-51定时器/计数器
- § 8-3 定时器/计数器的应用

MCS-51硬件结构简图

P53 图2-10

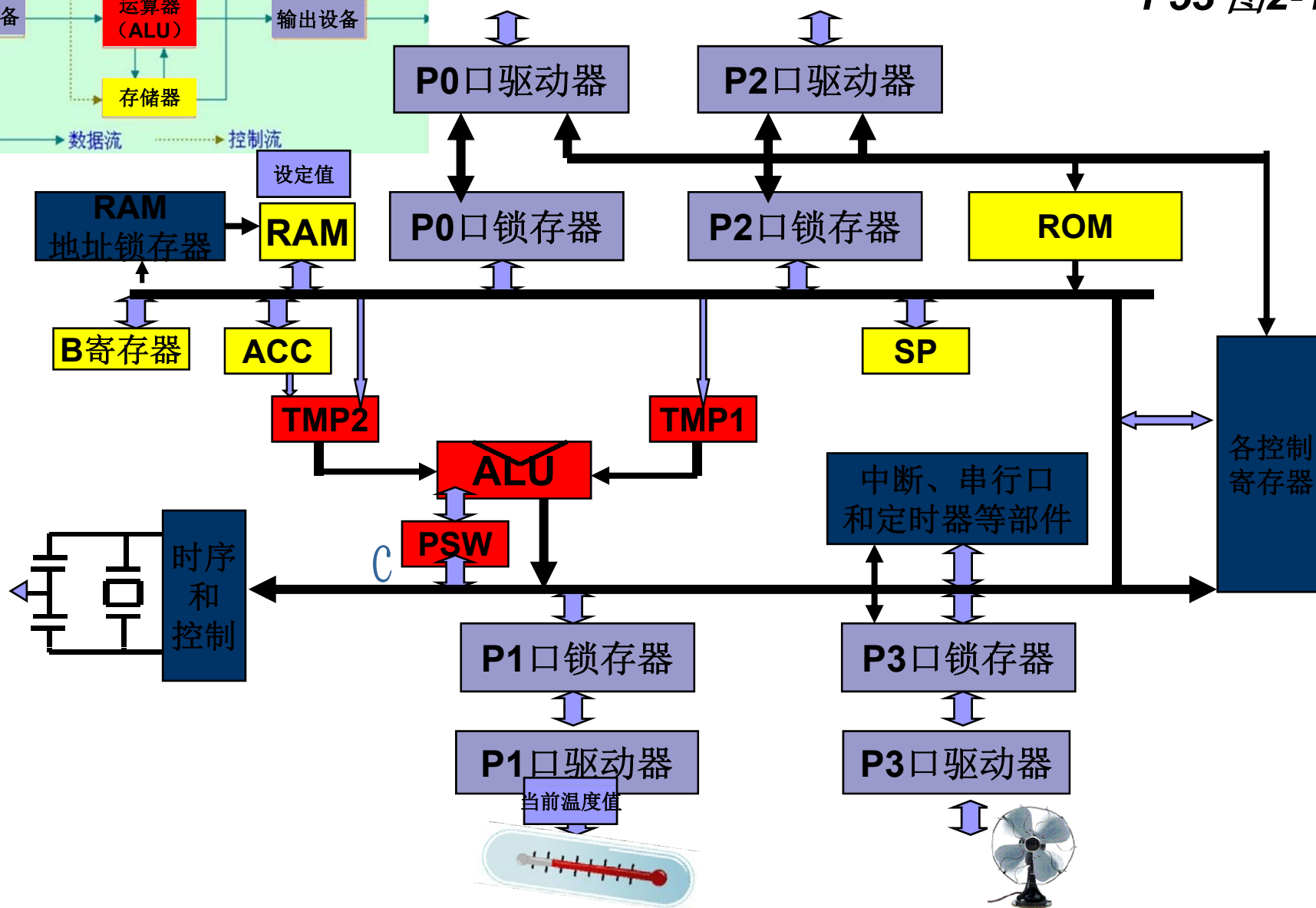


冯·诺依曼计算机结构

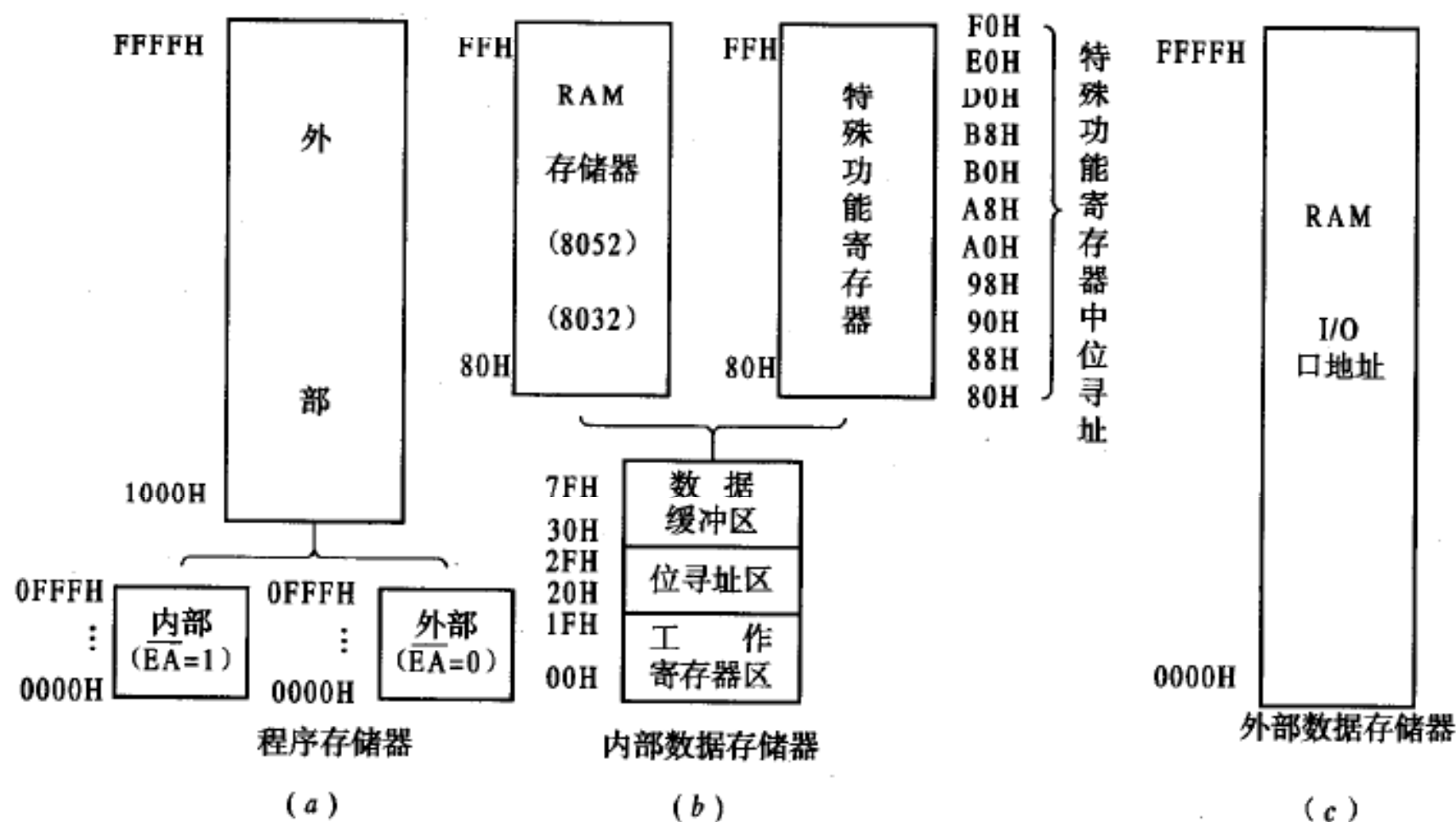


MCS-51硬件结构简图

P53 图2-10



MCS-51寄存器配置（物理空间）





■ 工作寄存器R0~R7

□ 00H~1FH

■ 位寻址区

□ 20H~2FH

□位地址为：00H~7FH

■ 用户 RAM 区

□ 30H~7FH

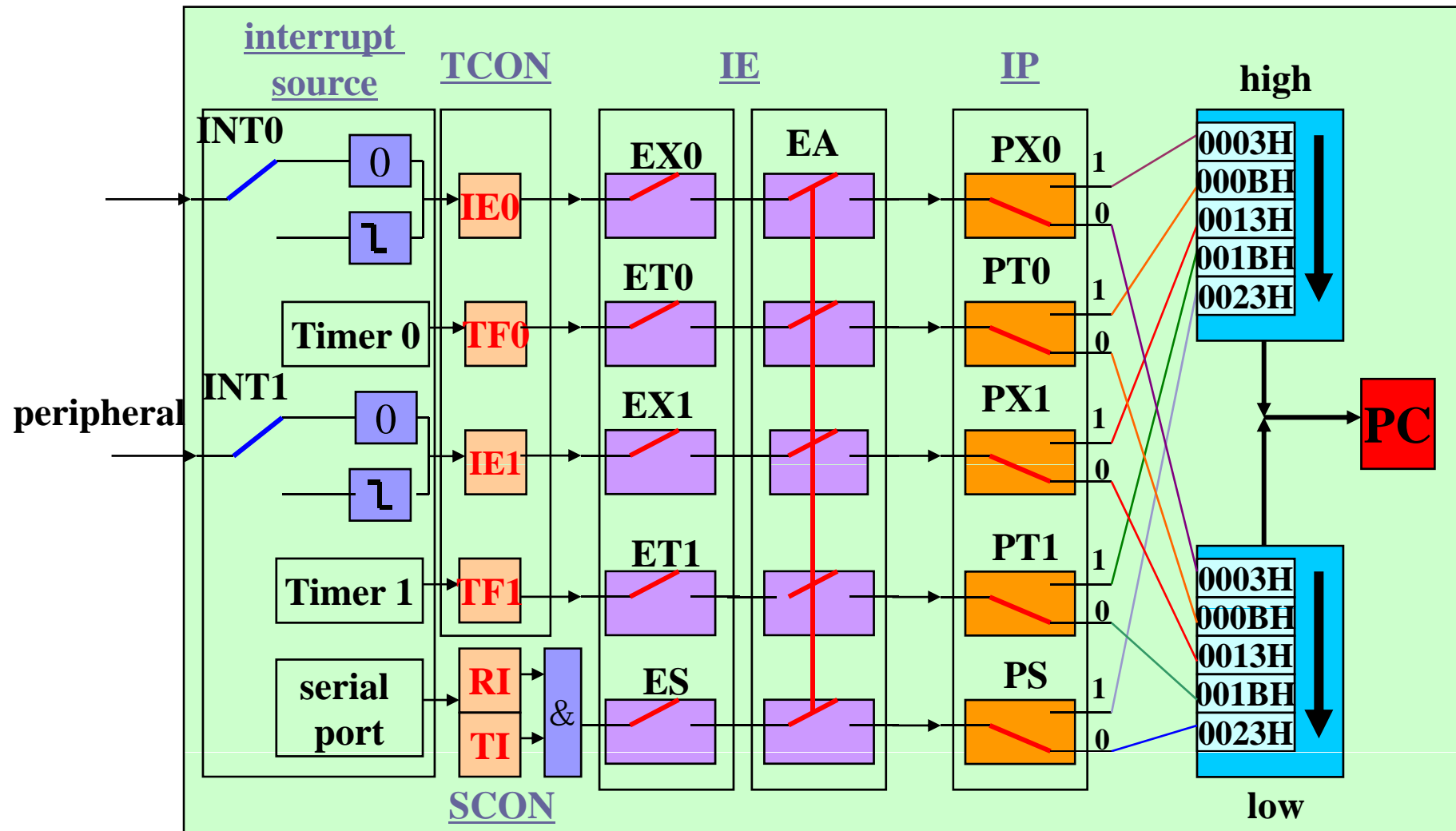
7FH									127
2FH	7F	7E	7D	7C	7B	7A	79	78	47
2EH	77	76	75	74	73	72	71	70	46
2DH	6F	6E	6D	6C	6B	6A	69	68	45
2CH	67	66	65	64	63	62	61	60	44
2BH	5F	5E	5D	5C	5B	5A	59	58	43
2AH	57	56	55	54	53	52	51	50	42
29H	4F	4E	4D	4C	4B	4A	49	48	41
28H	47	46	45	44	43	42	41	40	40
27H	3F	3E	3D	3C	3B	3A	39	38	39
26H	37	36	35	34	33	32	31	30	38
25H	2F	2E	2D	2C	2B	2A	29	28	37
24H	27	26	25	24	23	22	21	20	36
23H	1F	1E	1D	1C	1B	1A	19	18	35
22H	17	16	15	14	13	12	11	10	34
21H	0F	0E	0D	0C	0B	0A	09	08	33
20H	07	06	05	04	03	02	01	00	32
1FH									31
18H	寄存器区3								24
17H									23
10H	寄存器区2								16
0FH									15
08H	寄存器区1								8
07H									7
00H	寄存器区0								0

特殊寄存器SFR

- 占用字节地址：80H~FFH
- 位寻址寄存器：
 - 其字节地址可被8整除。
- 专用寄存器：
 - A、B、PSW、DPTR、SP
 - TMOD、TCON、SCON ...
- I/O接口寄存器：
 - P0、P1、P2、P3、SBUF

F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
DOH	07	06	05	04	03	02	01	00	PSW
	CY AC FD 05 06 07 08 09								
	TF2H TF2B CL K TC CL K H 1 H 2 TR 2								
	CY2 CP/RI2								
C8H	CF	CE	CD	CC	CB	CA	C9	C8	T2CON
	PT2 PS PT1 PSL PT0 P10								
B8H	—	—	BD	BC	BB	BA	B9	B8	IP
B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA ET2 EN ET1 EN ET0 EN EX								
A8H	AF	—	AD	AC	AB	AA	A9	A8	IE
A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
	SM0 SM1 SM2 SM3 SM4 SM5 SM6 SM7 TI RI								
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
	T0 T1 T2 T3 T4 T5 T6 T7								
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
80H	87	86	85	84	83	82	81	80	P0

Interrupt system structure chart





8-1 定时器/计数器概述


定时器、计数器是微机系统必需的部件，几乎所有的单片机均包含可编程定时器/计数器。

■ 定时器定义

□ 定时是对周期固定、已知的脉冲计数。

■ 计数器定义

□ 计数是对外界产生的周期不固定的未知脉冲计数。计数器的计数方式可以是加1计数，也可以是减1计数。



8-1 定时器/计数器概述

■ 定时器的应用

- 微机应用系统通常需要定时测量和控制
- 如：定时测量温度信号，定时输出电机的控制信号等

■ 计数器的应用

- 一些测控系统需要根据外部脉冲信号的计数结果来触发控制
- 如：某工件计数到一定数值后，发出打包信号，由机器进行自动打包



8-1 定时器/计数器概述

定时器的实现方法

■ 软件定时

- 通过执行一段循环程序来实现时间的延时
- 优点：无需额外的硬件，时间也比较准确
- 缺点：需要**CPU**运行程序，消耗**CPU**的时间

■ 硬件定时

- 用可编程定时器/计数器，对其编程实现
- 通过中断方法来完成定时功能或计数功能
- 优点：不占用**CPU**时间，应用非常广泛
- 单片机内部几乎都配备了定时器/计数器
(**Timer/Counter**)

8-1 定时器/计数器概述

■ 软件定时/计数举例

例1：编制一个延时**1ms**的子程序。

```
D2MS:  MOV  R7, #2
D2MS0: MOV  R6, #250
        DJNZ R6, $
        DJNZ R7, D2MS0
        RET
```

例2：统计开关按动次数，并存于**30H**单元。

```
MOV  30H, #0
UP:  JB  P1.0,$
      JNB P1.0,$
      INC 30H
      SJMP UP
```



8-1 定时器/计数器概述

- 可编程定时/计数器——可通过指令对以下几方面内容进行设置
 - 工作方式的选择：定时或计数
 - 预置定时或计数初值——当然计数的最大值是有一定限制的，这取决于计数器的位数。（计数的最大值也就限制了定时的最大值）
 - 设置是否允许中断请求（当定时时间到或计数终止时
 - 启动定时器或计数器工作



§ 8-2 MCS-51定时器/计数器

- 8-2-1 组成结构
- 8-2-2 工作原理
- 8-2-3 控制寄存器
- 8-2-4 工作方式



8-2-1 定时器/计数器组成结构

- **MCS51**单片机内有**2**个独立的**16**位的可编程定时器/计数器**T0**和**T1**
- 定时器/计数器**T0**、**T1**由以下几部分组成
 - 计数器**TH0**、**TL0**和**TH1**、**TL1**
 - 特殊功能寄存器**TMOD**、**TCON**
 - 时钟分频器
 - 内部总线
 - 输入引脚**T0**、**T1**



8-2-1 定时器/计数器组成结构

■ 计数器

- 8位计数器TH0、TL0和TH1、TL1分别构成T0和T1的16位计数器
- 它们都是以加”1”的方式完成计数

■ 特殊功能寄存器

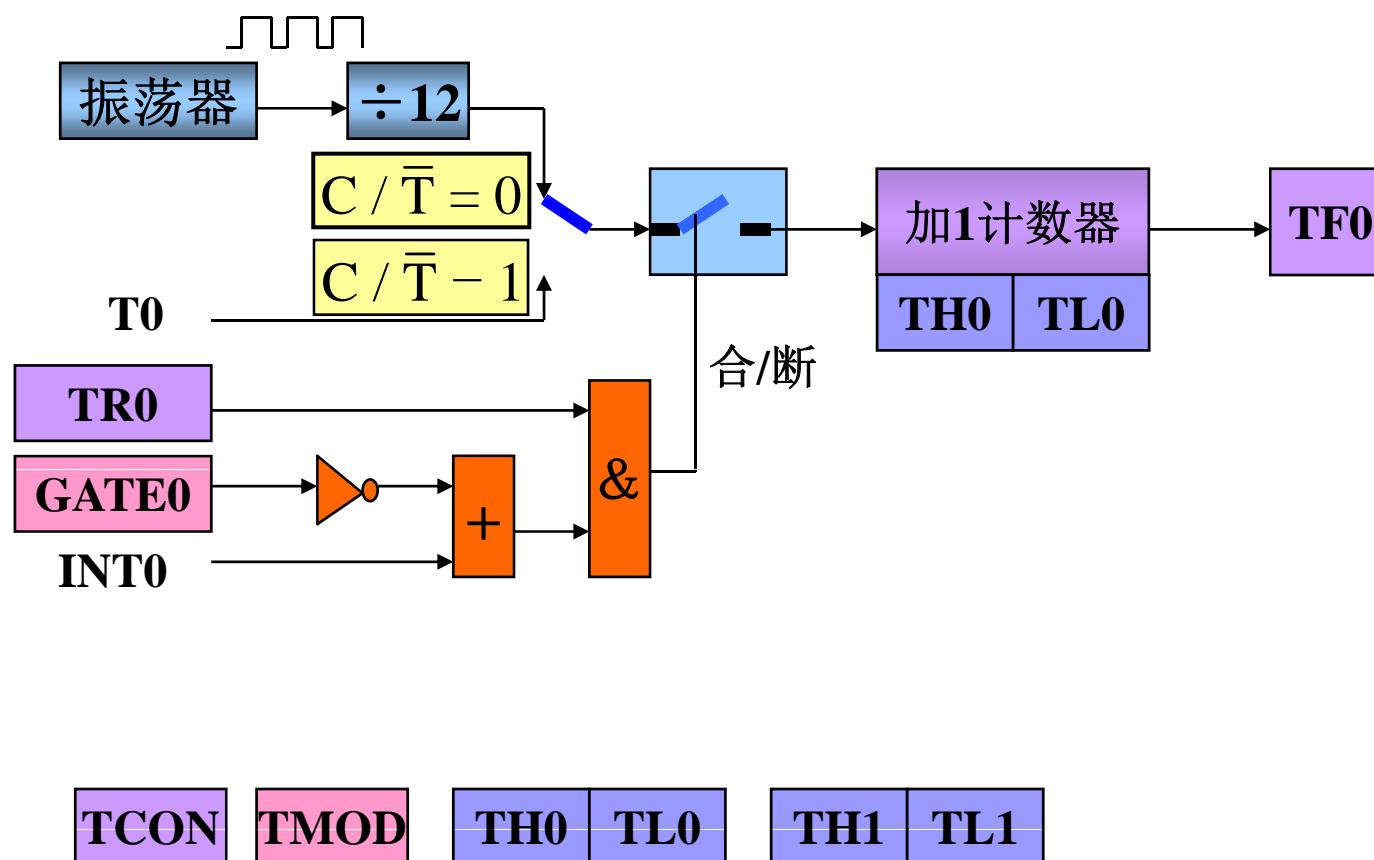
- TMOD: 控制定时/计数器的工作方式
- TCON: 控制定时/计数器的启动并记录T0、T1的溢出标志

■ 通过对TH0、TL0和TH1、TL1的初始化编程，可以预置T0、T1的计数初值

■ 通过对TMOD和TCON的初始化编程，可以写入方式字和控制字，控制T0、T1按规定的方式计数

8-2-1 定时器/计数器组成结构

T0的结构如图所示:





8-2-2 定时器/计数器工作原理

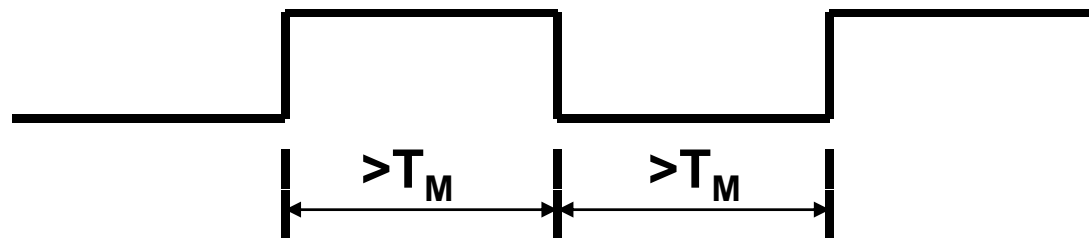
■ 定时器工作原理

- 定时工作方式时，计数器的输入脉冲为内部振荡器经过**12**分频后的输出(即机器周期的脉冲)
- 因此每个机器周期，计数器“**+1**”(即定时器的计数频率是振荡频率的**1/12**)
- 若单片机的晶振频率为**12MHz**，则计数频率为**1MHz**；**1μs**计数器“**+1**”
- 当计数器“**+1**”到溢出时，标志着定时时间到

8-2-2 定时器/计数器工作原理

■ 计数器工作原理

- 计数工作方式时，计数器的输入信号来自外部引脚 **T0(P3.4)**、**T1(P3.5)** 的脉冲
- 每输入一个脉冲，计数器 “+1”
- 实际工作时，**CPU** 在每个机器周期的 **S5P2** 采样外部输入引脚 **T0(T1)**，若一个机器周期的采样值为高电平，而下一个机器周期的采样值为低电平(即检测到一个下降沿)，则计数器 “+1”，完成一次计数操作。





8-2-2 定时器/计数器工作原理

■ 计数方式时，对外部脉冲的要求

- 因为是下降沿触发计数器+1，**CPU**是每个 T_M 检测一次外部引脚，因此检测到一个下降沿至少需要**2**个机器周期，所能计数的最高频率为单片机晶振频率的**1/24**
- 对外部脉冲的占空比没有特殊要求，但要求高电平和低电平的宽度至少为一个机器周期
- 外部脉冲电平要与**T0**、**T1**引脚电平匹配



8-2-2 定时器/计数器工作原理

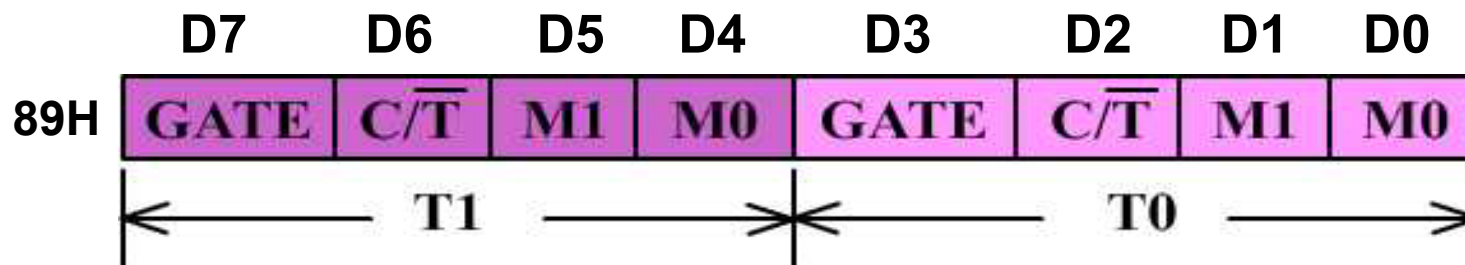
- 不管是定时方式还是计数方式，定时期**T0**和**T1**在工作时，均不占用**CPU**时间，只有当溢出时，才向**CPU**请求中断。因此，定时器/计数器是单片机内部效率高且工作灵活的部件。



8-2-3 控制寄存器

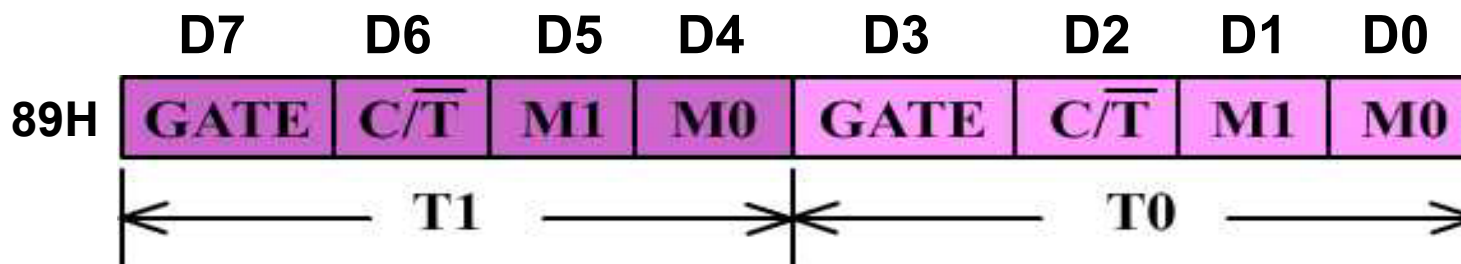
- 在应用定时器/计数器**T0**、**T1**前，首先要对其进行编程(初始化)，对**T0**、**T1**的编程是通过工作方式寄存器**TMOD**和控制寄存器**TCON**进行的。

方式寄存器 TMOD



TMOD: 控制定时/计数器的工作方式，字节地址为**89H**，只能按字节操作。单片机复位时，各位均被清“0”

方式寄存器 TMOD



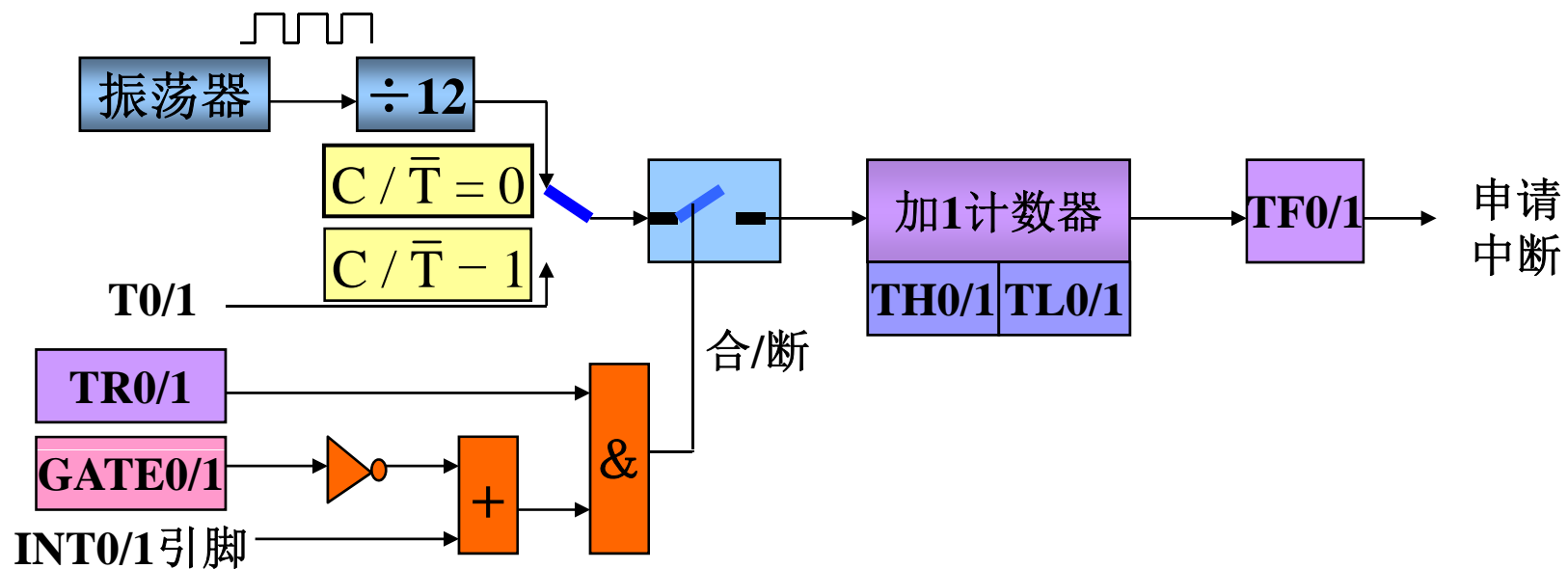
GATE：门控位。它对定时/数器的启动起着控制作用

- 当**GATE=1**时，定时计数器的启动除了受**TR0/1**控制外，还受**INT0/1**引脚的控制。 **结构图**

- ☐ 当**INT0**引脚为高电平时，**TR0**置位启动定时器**T0**
- ☐ 当**INT1**引脚为高电平时，**TR1**置位启动定时器**T1**

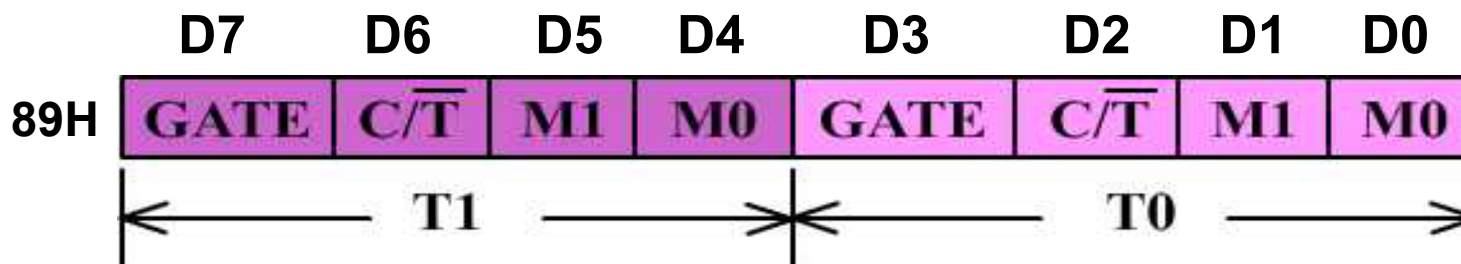
- 当**GATE=0**时，外部中断引脚**INT0/1**不参与控制，此时只要置位**TR0/1**，相应的计数器就开始计数，而不管引脚**INT0/1**的电平是高还是低。

定时器/计数器结构



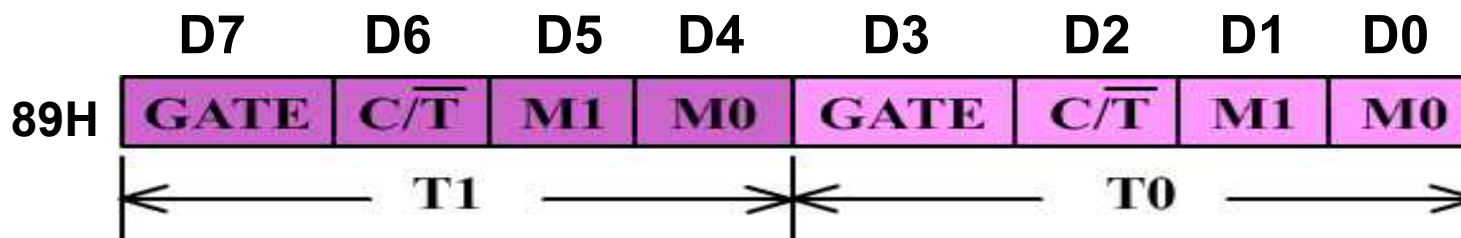
返回

方式寄存器 TMOD



- C / \bar{T} ：功能选择位。
- 用来确定T0(T1)是工作在计数方式还是工作在定时方式
 - C / \bar{T} = 0 为定时方式，C / \bar{T} = 1 为计数方式
 - 外部引脚上输入的每一个脉冲的负跳变使计数值加1
 - 最小的计数周期为两个机器周期。例如，若单片机晶振频率为12MHZ，则外部计数脉冲的最高频率只能为500KHZ

方式寄存器 TMOD



- **M1和M0**：确定**T0(T1)**的具体工作模式。**M1、M0**的四种组合刚好与四种工作模式对应(方式0、方式1、方式2、方式3)


M1 M0	操作方式	功能
0 0	方式0	13位计数器
0 1	方式1	16位计数器
1 0	方式2	可自动重装载的8位计数器
1 1	方式3	T0分为两个独立的8位计数器， T1停止计数

TCON

8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TCON: 88H, 可以按字节操作, 也可以按位操作。

- **TF0/1**: **T0/1**溢出中断标志位。当**T0**计数溢出时, **TF0=1**。在允许中断的情况下, **CPU**响应**T0**中断, 转向**T0**中断服务程序, 此时由硬件自动将**TF0**清**0**。也可由程序查询后清“**0**”
- **TR0/1**: 为**T0/1**启动控制位。当**TR0=1**时, 启动**T0**; **TR0=0**时, 关闭**T0**。该位由软件进行设置。
- **IE0/1**: 外部中断的中断请求标志位。
- **IT0/1**: 外部中断的触发方式选择位。



计数寄存器

- 单片机内部有两个16位的定时/计数器T0和T1。
每个定时/计数器占用两个特殊功能寄存器：
 - T0由TH0和TL0两个8位计数器组成，字节地址分别是8CH和8AH。
 - T1由TH1和TL1两个8位计数器组成，字节地址分别是8DH和8BH。
 - 用于存放定时或计数的初值。当计数器工作时，其值随计数脉冲做加1变化。
 - 复位后，这四个寄存器全部清“0”。

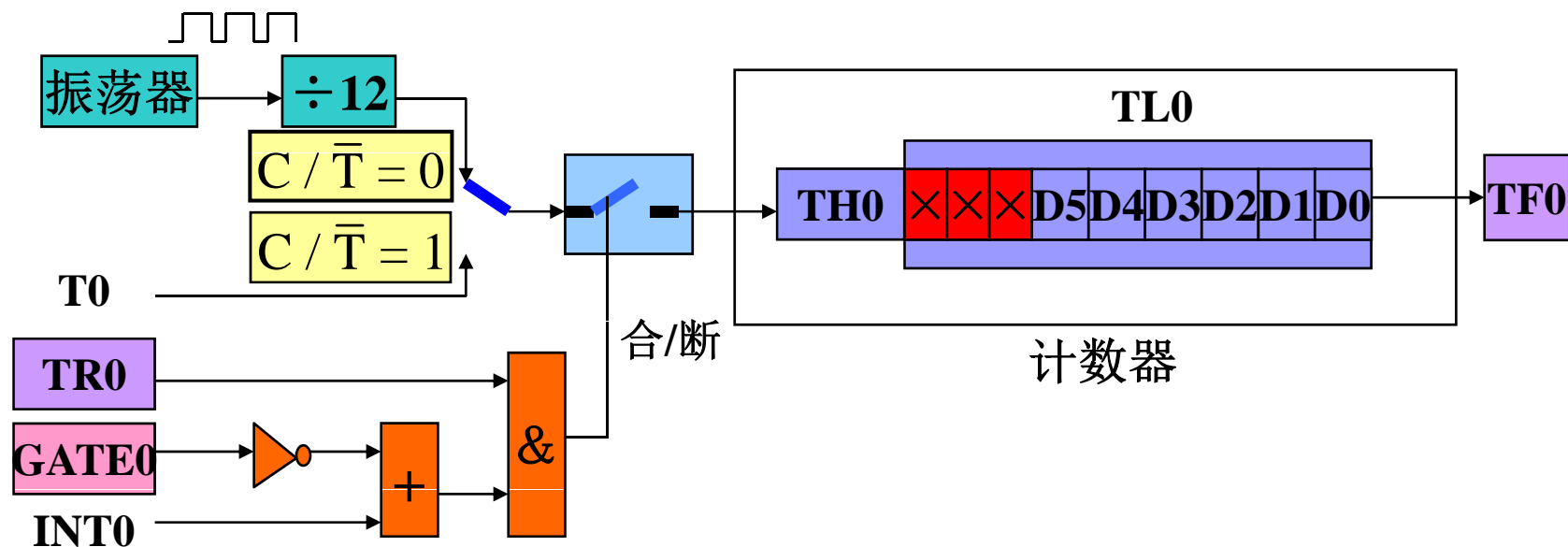
8-2-4 定时器/计数器的工作方式

- **M1和M0**: 确定**T0(T1)**的具体工作模式。**M1、M0**的四种组合刚好与四种工作模式对应(**方式0**、**方式1**、**方式2**、**方式3**)

M1 M0	操作方式	功能
0 0	<u>方式0</u>	13位计数器
0 1	<u>方式1</u>	16位计数器
1 0	<u>方式2</u>	可自动重载的 8位计数器
1 1	<u>方式3</u>	T0 分为两个独立的 8位计数器 , T1 停止计数

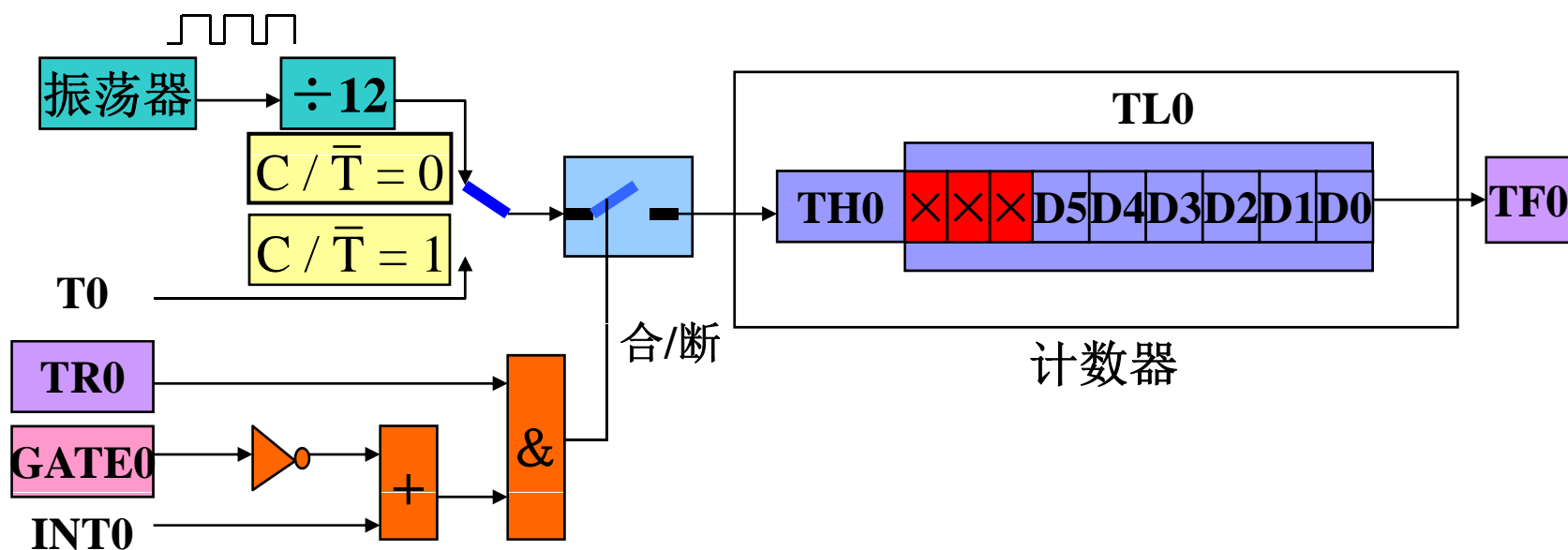
方式0 (以T0为例)

- 在方式0下，T0构成一个13位的计数器，由TH0的8位和TL0的低5位组成，TL0的高3位未用，满计数值为 2^{13} 。
- T0启动后立即加1计数，当TL0的低5位计数溢出时向TH0进位，TH0计数溢出则对相应的溢出标志位TF0置位，以此作为定时器溢出中断标志。当单片机进入中断服务程序时，由内部硬件自动清除该标志。



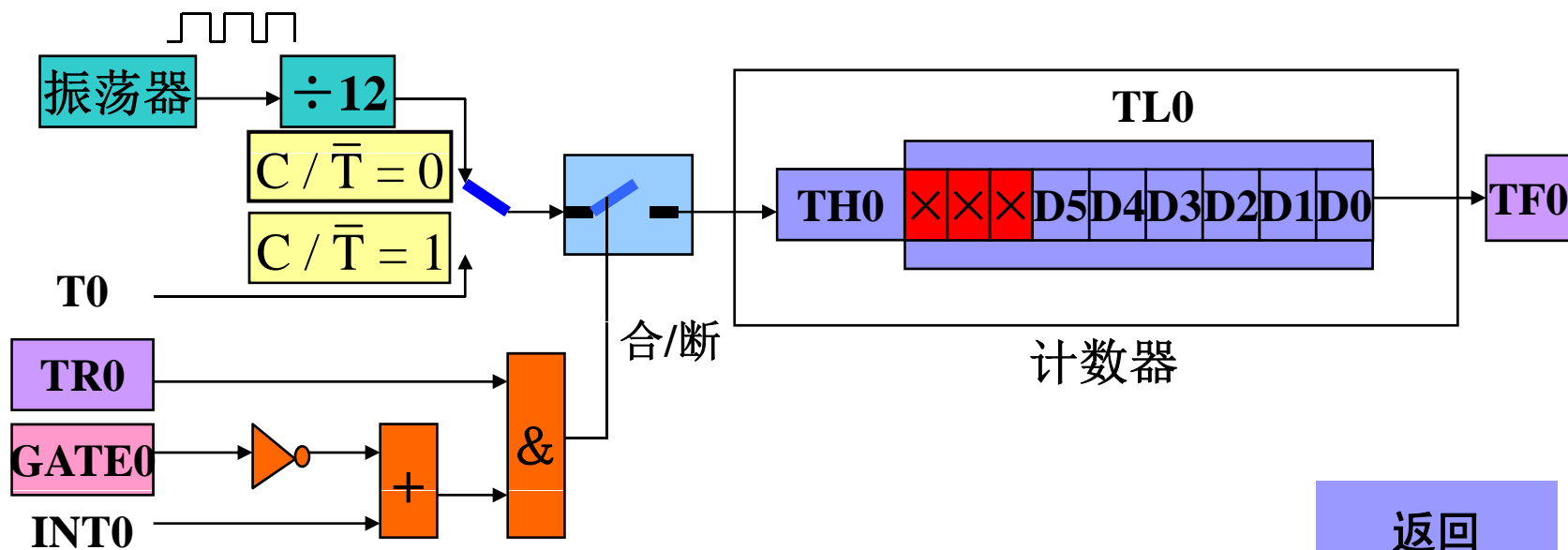
方式0

- **GATE**是门控位，用来释放或封锁**INT0/1**信号，只有当**GATE=1**时，**INT0/1**的状态才起作用。
 - **GATE=0**,开关**K**的控制信号=**TR0**，即开关只受**TR0**的控制。
 - **TR0=0**，**K**断开，计数器不计数
 - **TR0=1**，**K**闭合，计数器开始计数



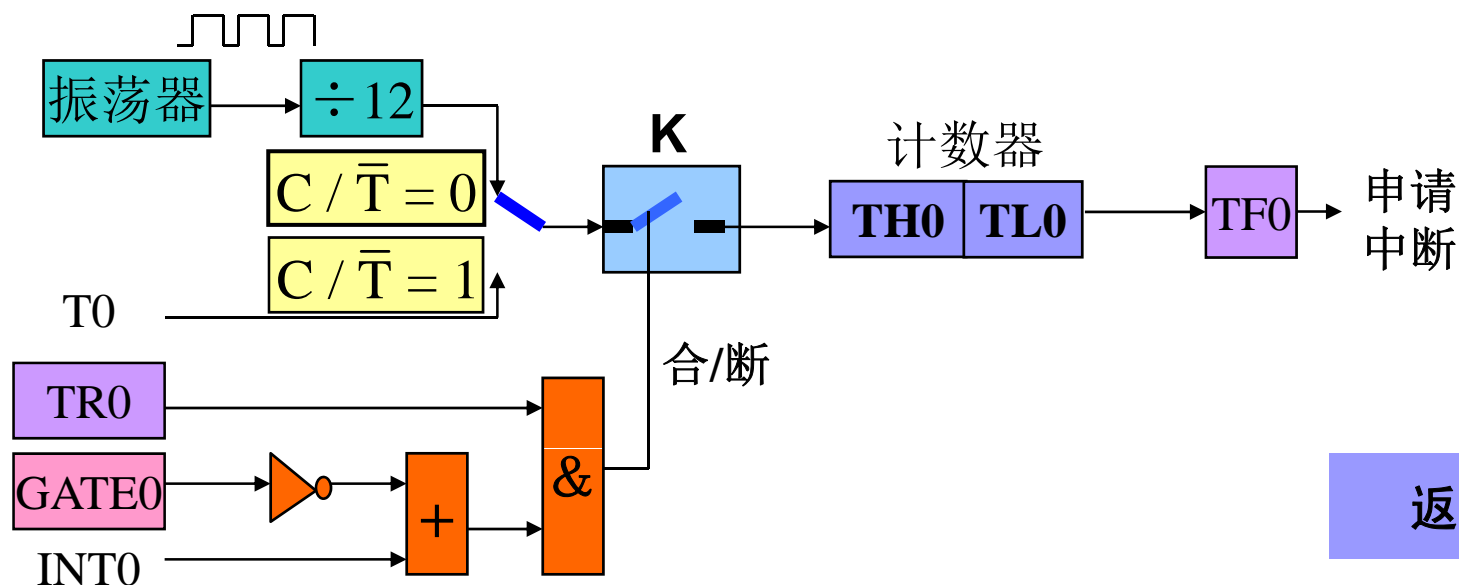
方式0

- 当**GATE=1**和**TR0/1=1**时，则计数器的启动受外部中断信号**INT0/1**的控制
- 当**INT0/1**为高电平时，计数器开始计数
- 当**INT0/1**为低电平时，停止计数。
- 利用这一功能可测量**INT0**引脚上正脉冲的宽度。
- **TF0**是定时器/计数器**T0**溢出标志。



方式1(以T0为例)

- T0构成一个16位的计数器，由TH0的8位和TL0的8位组成，满计数值为 2^{16} 。
- T0启动后立即加1计数，当TL0计数溢出时向TH0进位，TH0计数溢出则对相应的溢出标志位TF0置位，以此作为定时器溢出中断标志。
- 当单片机进入中断服务程序时，由内部硬件自动清除该标志。



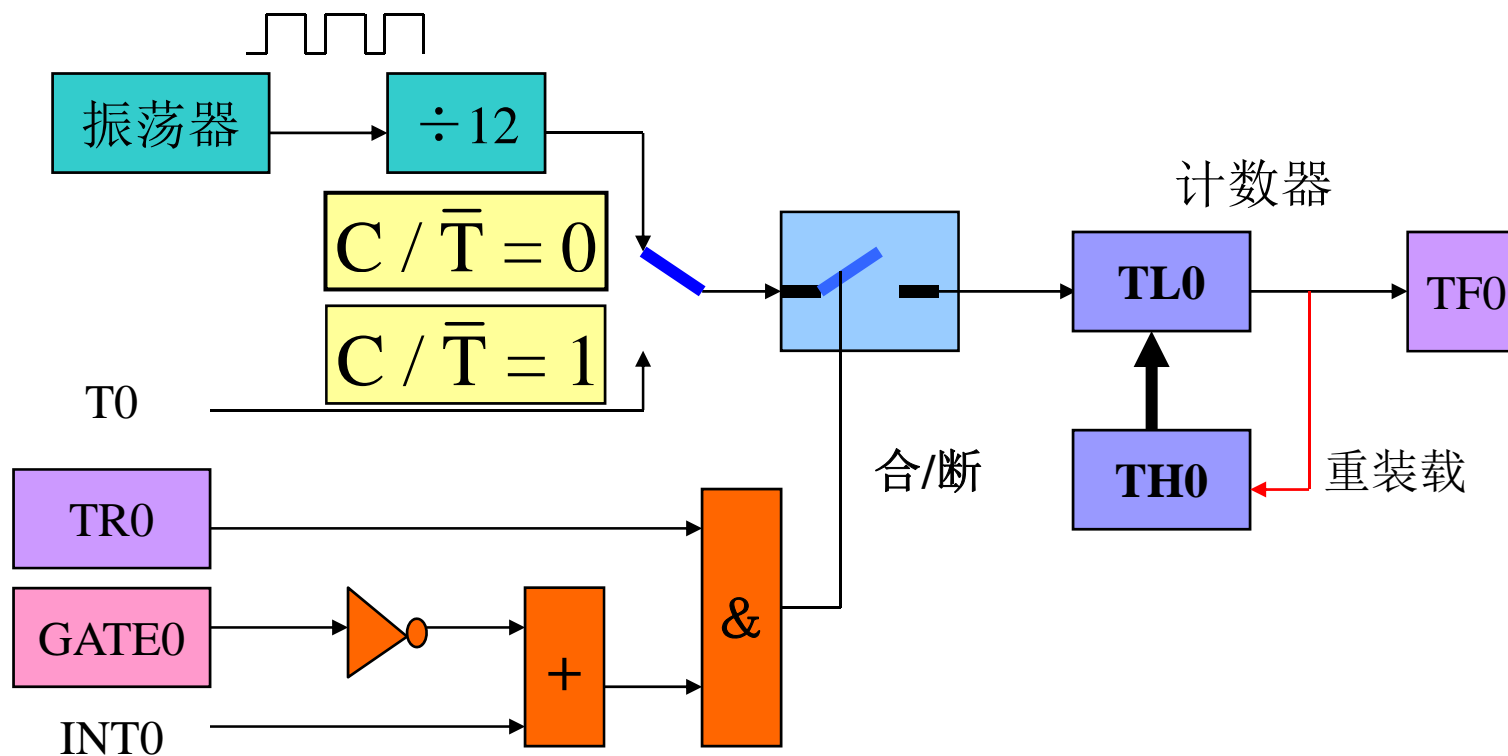
返回



方式2 (以T0为例)

- TH0和TL0被当作两个8位计数器，计数过程中，TH0寄存8位初值并保持不变，由TL0进行8位计数。当低8位计数溢出时，除了可产生中断申请外，还将TH0中保存的内容向TL0重新装入，以便于重新计数，而TH0中的初值仍然保留，以便下次再行对TL0进行重装。
- 方式2对于连续计数比较有利。这时不需要在溢出后用软件重新装入计数初值，而是可以自动装入，但此时计数的长度将受到很大的限制，只有 $2^8=256$ 次。

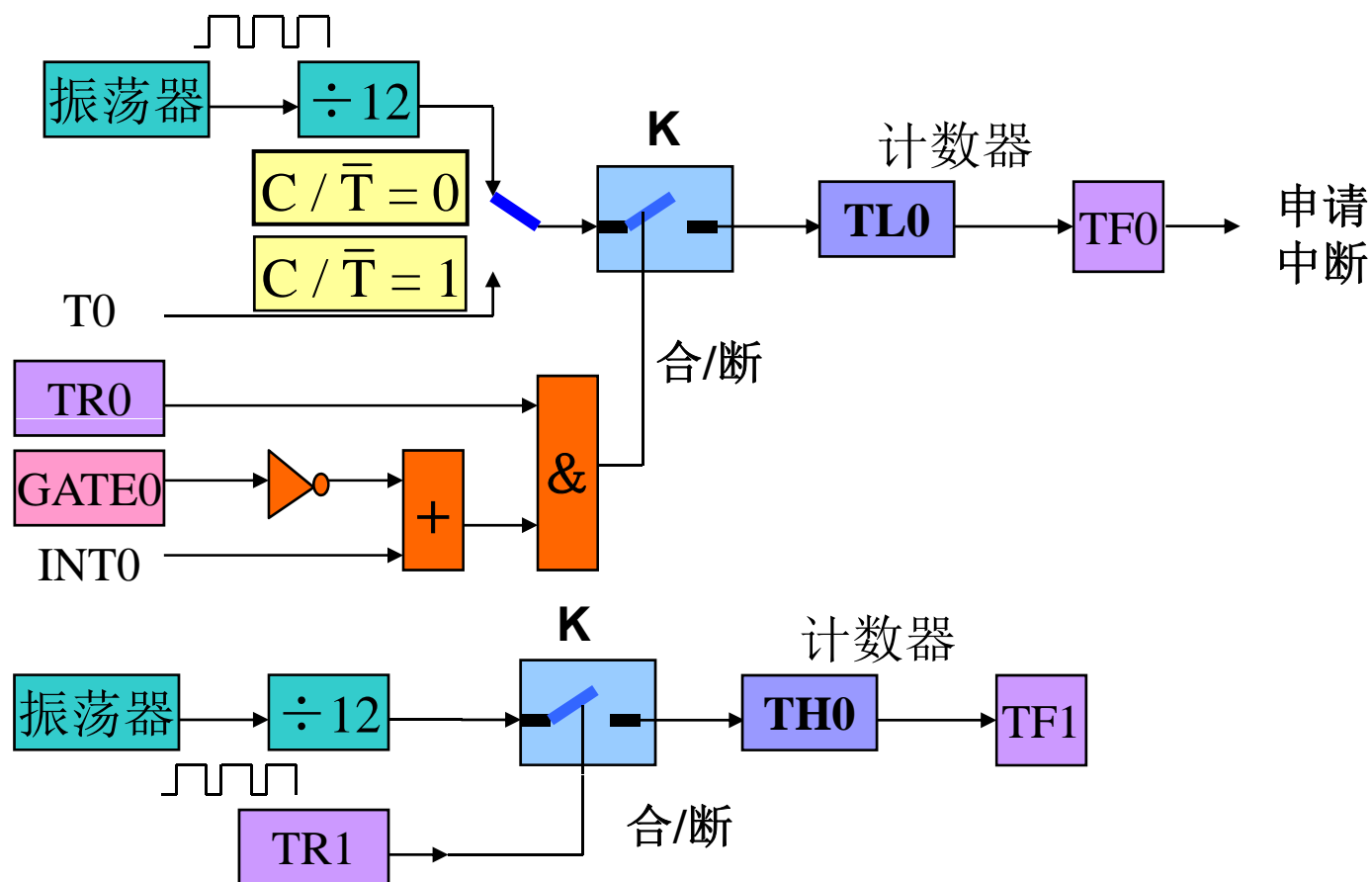
方式2结构逻辑图



返回

方式3

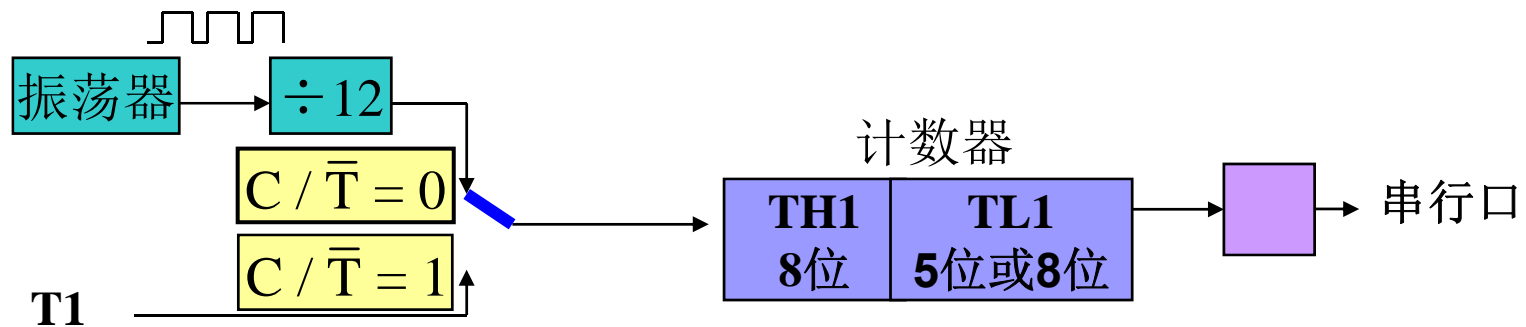
- 方式3只适用于定时/计数器T0。这种工作模式下，定时/计数器T0被拆成2个独立的定时/计数器来用。



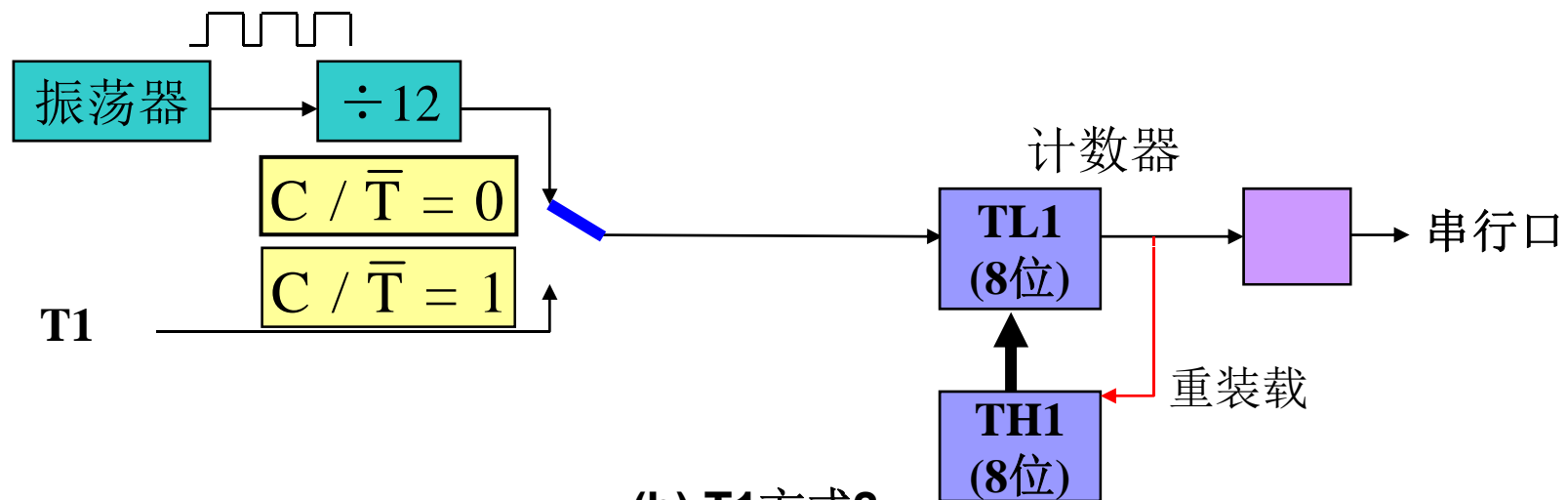


- 在**T0**用作方式**3**时，**T1**可设置为方式**0~2**
- 此时，通过**C/T**切换**T1**的定时器或计数器工作方式，计数溢出时，只能将输出送入串行口。设置好工作方式后，**T1**自动开始运行，若要停止操作，只需设置**T1**为方式**3**即可。因此通常将定时器**T1**设置为方式**2**，用作波特率发生器。
- 方式**3**只适用于定时器**T0**，定时器**T1**无方式**3**，如果设置**T1**位方式**3**，**T1**就停止工作。

工作方式结构逻辑图



(a) T1方式1或0




(b) T1方式2




§ 8-3 定时器/计数器的应用

- 8-3-1 定时器/计数器初始化
- 8-3-2 应用举例



8-3-1 定时器/计数器初始化

- 可编程定时器/计数器的初始化，包含以下几个步骤：
 - 确定工作方式——对方式寄存器**TMOD**赋值
 - 预置定时初值或计数初值，将初值写入**TL0**、**TH0**或**TL1**、**TH1**
 - 中断设置(给**IE**有关位赋值)，以开放或禁止定时器/计数器中断(**ET0**、**ET1**、**EA**)
 - 启动定时器/计数器，令**TCON**中的**TR0**或**TR1**为“1”



8-3-1 定时器/计数器初始化


在初始化过程中，要置入定时值或计数值的初值，这时要作一些计算。方法如下：

设计数器的最大值为 M (在不同的工作方式的， $M=2^L$, L 与工作方式有关。

方式0: $M=2^{13}=8192$;最长定时时间 $8192 \times T_M$ 。

方式1: $M=2^{16}$ 或 $=65536$;最长定时时间 $65536 \times T_M$ 。

方式2、3: $M=2^8=256$ ，最长定时时间 $256 \times T_M$ 。



8-3-1 定时器/计数器初始化

定时或计数置入的初值X可这样来计算：

计数方式时：计数值= $M - X$
(X即为计数值的补码)

定时方式时：定时值 $t(\mu s) = (M - X) \cdot T_M = (M - X) \cdot 12 / f_{osc}$ ，
所以 $X = M - t(\mu s) \times f_{osc}(\text{MHz}) / 12$ 。

其中 T_M 为计数周期，它是单片机时钟周期的12倍。

$1/t$ 称为溢出率。

8-3-1 定时器/计数器初始化

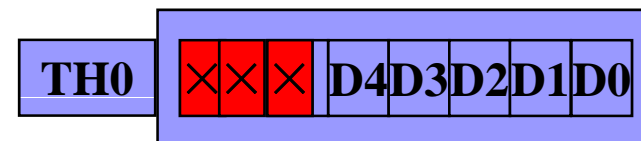
当时钟周期为 $1 / 12\text{MHz}$ 时，计数周期为 $1\mu\text{s}$ 。若定时器工作在方式0，则最大定时值为 $2^{13} \times (1\mu\text{s}) = 8.192\text{ms}$ ；若工作在方式1，则最大定时值为 $2^{16} \times (1\mu\text{s}) = 65.536\text{ms}$

例 若单片机的频率为 12MHz ，请计算 2ms 所需要的定时器初值。

解：计数脉冲个数为 $2/0.001=2000$

若方式0，则计数初值为： $2^{13}-2000=6192=1830\text{H}$ TL0

TH0=C1H, TL0=10H



若方式1，则计数初值为： $2^{16}-2000=63536=\text{F830H}$

TH0=F8H, TL0=30H

8-3-2 定时器/计数器应用

■ 作定时器用

□ 定时器方式0的应用

例：设主频为**12MHz**，利用定时器**T1**定时，使**P1.0**输出周期为**2ms**的方波。

解：周期为**2ms**的方波即每**1ms**改变一次电平，故定时值应为**1ms**，采用方式**0**，即**13**位计数器。

定时初值： **$X = M - \text{计数次数}$**

$$= 8192 - 1000 = 7192 = 1C18H$$

写入**TH1**、**TL1**的计数值应扩展为**16**位，**TL1**的高**3**位置**0**。

1110000000011000B = E018; **TH1**初值为**E0H**，**TL1**初值为**18H**

GATE	C / \bar{T}	M1	M0				
0	0	0	0				

8-3-2 定时/计数器应用

编程:

```
ORG 0000H
LJMP MAIN
ORG 001BH           ;T1中断入口
AJMP BRT1           ;转T1中断服务
ORG 0100H
MAIN: MOV TMOD, #0   ;T1按方式0工作
      MOV TH1, #0EH
      MOV TL1, #18H  ;给计数器赋值
      SETB EA        ;CPU开中断
      SETB ET1       ;T1允许中断
      SETB TR1       ;启动T1
      SJMP $         ;模拟主程序
ORG 0800H
BRT1: MOV TH1, #0EH
      MOV TL1, #18H  ;重赋T1初值
      CPL P1.0       ;输出方波
      RETI           ;返回
```

8-3-2 定时/计数器应用

C51程序:

```
#include <reg51.h>
sbit P10=P1^0;
void main()
{
    TMOD=0;                //T1按方式0工作
    TH1=7192/32;
    TL1=7192%32;           //给计数器赋值
    EA=1;                  //CPU开中断
    ET1=1;                 //T1允许中断
    TR1=1;                 //启动T1
    while(1);              //模拟主程序
}

void timer1() interrupt 3 using 1
{
    TH1=7192/32;
    TL1=7192%32;           //重赋T1初值
    P10=~P10;              //输出方波
}
```

8-3-2 定时/计数器应用

例：根据上例要求产生周期为**2ms**方波，但不用中断方式，而用查询方式工作，查询标志为**TF1**。

解：利用方式**1**，**16**位计数器，当定时时间到，**T1**计数器溢出使**TF1**置“**1**”，由于不采用中断方式，**TF1**置“**1**”后不会自动复“**0**”，而需要指令将**TF1**清“**0**”。

编程：

```
MOV    TMOD, #10H    ;置T1位方式1
SETB   TR1           ;启动T1定时
LOOP:  MOV    TH1,    #0FCH
        MOV    TL1,    #18H    ;装入初值
        JNB    TF1,    $        ;TF1=0,则继续查询
        CPL    P1.0          ;输出方波
        CLR    TF1           ;0-TF1
        SJMP   LOOP          ;重复下一个循环
```

程序很简单，但是要占用**CPU**时间，使**CPU**效率不高

8-3-2 定时/计数器应用

假如要求延时1秒，用16位计数器是不够的，而需要多次定时，比如定时一次为50ms，就需要20次。

编程：

```
MOV    R7,    #14H    ;设置定时次数
MOV    TMOD, #10H    ;启动T1定时
LOOP:  MOV    TH1,    #3CH
        MOV    TL1,    #0B0H ;装入初值
        SETB   TR1
HERE:  JNB    TF1,    HERE
        CLR    TF1
        DJNZ   R7,    LOOP ;1s延时时间未到，继续
        .....          ;此时，1s延时到
```

再次循环时要注意初值的重置，否则时间不对

仅仅利用定时器产生定时而不用中断，仍要占用CPU时间，使CPU无法做其他事情，不是好的定时方法。

另用定时中断，使CPU腾出时间从事其他工作，才是比较好的定时方法。

8-3-2 定时/计数器应用


定时与中断结合产生延时。

要求定时**2秒**，由定时器产生**50ms**定时并产生中断。

经**40**次中断后，即为**2秒**。

主程序：

```
MOV  TMOD,#10H  ;启动T1定时
MOV  TH1,  #3CH
MOV  TL1,  #0B0H      ;装入初值
MOV  R7,   #28H
SETB EA
SETB ET1
SETB TR1
.....
SJMP $
```



8-3-2 定时/计数器应用

中断程序:

```
ORG 001BH  
LJMP T1INT
```

```
ORG 0030H  
T1INT:MOV TH1, #3CH  
MOV TL1, #0B0H  
DJNZ R7, AGAIN  
MOV R7, #28H  
.....  
延时2秒到处理程序  
AGAIN: RET1
```

8-3-2 定时/计数器应用

作定时器用

定时器方式**2**的应用

要求在输入信号发生负跳变时，从**P1.0**输出一个**250μs**的负脉冲。

T0作计数器，记录输入脉冲，用方式**2**，计数初值为**FFH**

T1作定时器，定时**250μs**，用方式**2**，初值为**6**。
($T_M=1\ \mu s$)

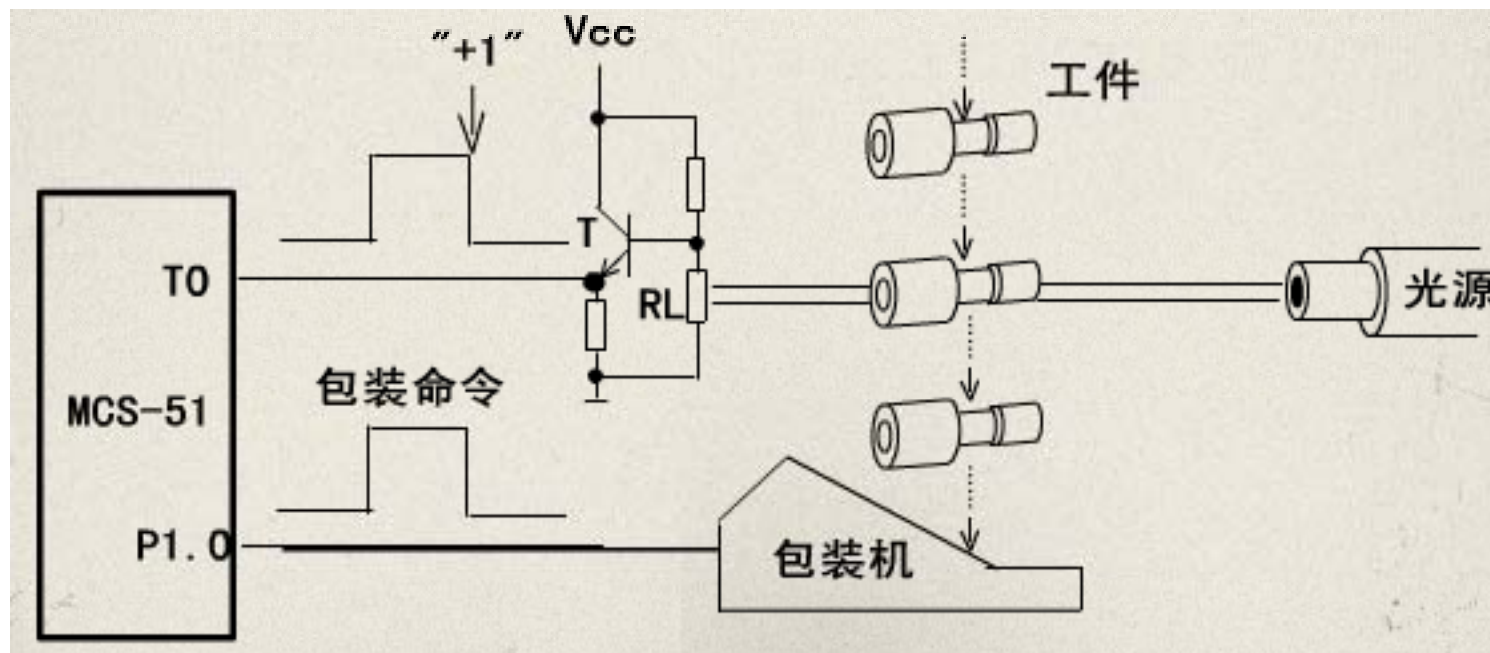
```
MOV    TMOD, #26H
MOV    TH1,   #06H
MOV    TL1,   #06H
MOV    TH0,   #0FFH
MOV    TL0,   #0FFH
LOOP:  SETB    TR0
        JNB    TF0,   $
        CLR    TF0
        CLR    TR0
        CLR    P1.0
        SETB   TR1
        JNB    TF1,   $
        SETB   P1.0
        CLR    TF1
        CLR    TR1
        SJMP   LOOP
```


8-3-2 定时/计数器应用

■ 作计数器用

例：用**T0**计数一生产流水线上加工好的工件，每次计数到**100**个时，要求系统发出一包装命令，命令包装机打包。要求记录每天生产的工件箱数。

解：硬件电路如下图所示



8-3-2 定时/计数器应用

RL为光敏电阻，当有工件通过时，**RL**阻值升高，三极管输出高电平，则每通过一个工件，会产生一个高脉冲信号。**T0**作计数器，用工作方式**2**。

1)方式字**TMOD: 06H**

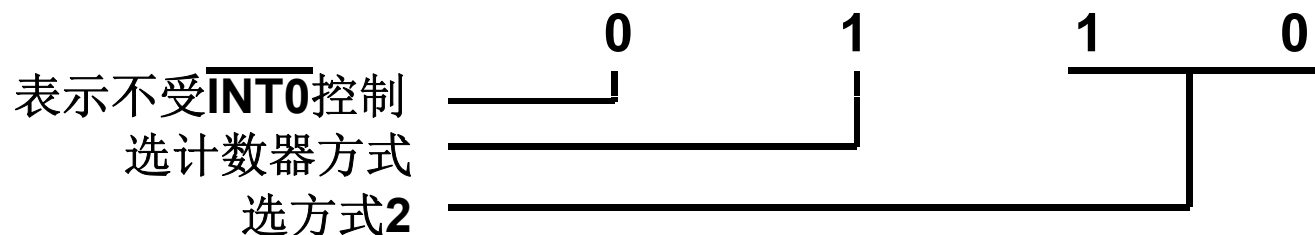
2)计数初值 **$X=256-100=156 \Rightarrow 9CH$**

3)用**P1.0**输出打包机的包装命令(高脉冲)

4)用**R5R4**作每天生产工件箱数的计数器

方式字**TMOD**

				GATE	C / \bar{T}	M1	M0
--	--	--	--	------	---------------	----	----



8-3-2 定时/计数器应用

编程:

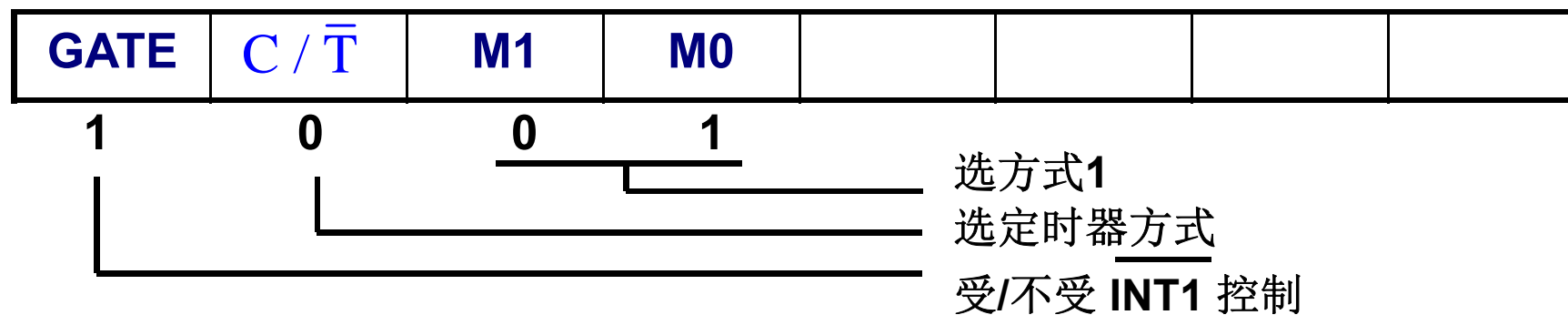
```
ORG 0000H
SJMP MAIN
ORG 000BH           ;T0中断入口
LJMP COUNT          ;转向中断服务
ORG 0040H
MAIN: CLR P1.0       ;P1.0为低
      MOV R5, #0
      MOV R4, #0      ;箱数计数器清“0”
      MOV TMOD, #6    ;置T0工作方式
      MOV TH0, #9CH   ;置重载系数
      MOV TL0, #9CH   ;置计数初值
      SETB EA         ;CPU开中断
      SETB ET0        ;T0开中断
      SETB TR0        ;启动T0
      SJMP $          ;模拟主程序
```

8-3-2 定时/计数器应用

```
COUNT:    ORG  0800H
           MOV  A,   R4
           ADD  A,   #1
           MOV  R4,   A
           MOV  A,   R5
           ADDC A,   #0
           MOV  R5,   A      ;箱计数器加“1”
           SETB P1.0        ;输出包装机打包信号
           MOV  R3,   #10
DLY:       NOP
           DJNZ R3,   DIY    ;高脉冲有一定宽度
           CLR  P1.0        ;停止包装
           RETI             ;中断返回
```

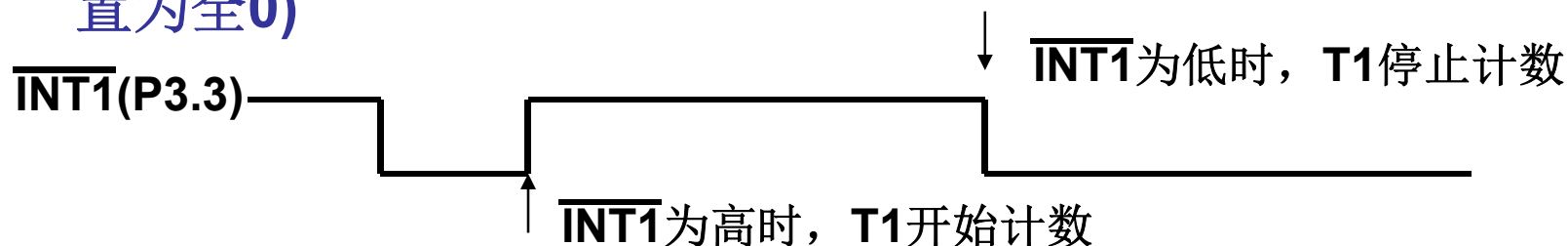
8-3-2 定时/计数器应用

■ 门控位GATE的应用



例：用GATE控制位，测量INT1(P3.3)引脚上的正脉冲的宽度(设晶振为12MHz，正脉冲宽度小于65ms)。

测量时，在INT1端为“0”时置TR1为“1”，由INT1端变“1”来启动计数，INT1端再次变成“0”时停止工作，这时的计数值就是被测正脉冲的宽度。(能测到的最长脉冲宽度=65535 μ s，初始设置为全0)



8-3-2 定时/计数器应用

```
MOV  TMOD, #90H ;设置方式控制字
CLR  TR1
MOV  TL1,    #0
MOV  TH1,    #0 ;T1从0开始计数, 计TM的个数
JB   P3.3,   $ ;等INT1变为低电平
SETB TR1     ;T1允许计数
JNB  P3.3,   $ ;等INT1变高电平
JB   P3.3,   $ ;等INT1再变低电平
CLR  TR1     ;停止计数
MOV  A,      TL1 ;累计的TM个数, 即脉冲宽度
MOV  B,      TH1 ;μs数
```

INT1:



TR1="1"



8-3-2 定时/计数器应用

例：用定时器**T1**定时，完成日历时钟秒、分、时的定时。设晶振频率为**12MHz**。

解：首先完成**1秒**定时。在此基础上，每计满**60秒**，分钟加**1**；每满**60分**，时加**1**；计满**24小时**，时钟清零，然后从**0时**继续重复上述过程。因此，要完成日历时钟的设计，首先解决**1秒**的定时。**51**单片机方式**1**的定时时间最长，定时时间的最大值**T_{MAX}**为：

$$T_{MAX} = M \times 12 / f_{osc} = 65536 \times 12 / (12 \times 10^6 \text{Hz}) = 65536 \mu\text{s} = 65.536 \text{ms}$$

显然不能满足**1秒**的定时要求，因此需要设置一个软件计数器，对分、时的计数同样通过软件完成。在此采用片内**50H**、**51H**、**52H**、**53H**单元分别作为**50ms**、秒、分、时的计数单元

设置**T1**定时**50ms**，此时**T1**的初始值**X**为：

$$(M-X) \times 1 \times 10^{-6} = 50 \times 10^{-3}$$

$$X = 65536 - 50000 = 15536 = 3CB0\text{H}$$

8-3-2 定时/计数器应用

```
CLR  A
MOV  50H,  A      ;50ms个数清0
MOV  51H,  A      ;秒清0
MOV  52H,  A      ;分清0
MOV  53H,  A      ;时清0
MOV  TMOD, #10H ;设定时器1为方式1
MOV  TH1,  #3CH ;赋初值
MOV  TL1,  #0B0H
SETB TR1          ;启动T1
L2:  JBC  TF1,  L1  ;查询计数溢出标志TF1
     SJMP L2
```


8-3-2 定时/计数器应用

```
L1:    MOV    TH1,    #3CH
        MOV    TL1,    #0B0H
        INC    50H                ;50ms个数加1
        MOV    A,      50H
        CJNE   A,      #20,    L2    ;判断是否到1秒
        MOV    50H,    #00H        ;到1秒, 50ms个数清0
        INC    51H                ;秒个数+1
        MOV    A,      51H
        CJNE   A,      #60,    L2    ;判断是否到60秒
        MOV    51H,    #00H        ;到60秒, 秒个数清0
        INC    52H                ;分个数+1
        MOV    A,      52H
        CJNE   A,      #60,    L2    ;判断是否到60分
        MOV    52H,    #00H        ;到60分, 分个数清0
        INC    53H                ;分个数+1
        MOV    A,      53H
        CJNE   A,      #24,    L2    ;判断是否到24小时
        MOV    53H,    #00H        ;到24小时, 小时个数清0
        SJMP   L2                ;反复循环
```